A General Approach to Reasoning with Probabilities

Federico Cerutti[a], Matthias Thimm[b]

[a] *Cardiff University, UK*
[b] *University of Koblenz-Landau, Germany*

Abstract

We propose a general scheme for adding probabilistic reasoning capabilities to a wide variety of knowledge representation formalisms and we study its properties. Syntactically, we consider adding probabilities to the formulas of a given base logic. Semantically, we define a probability distribution over the subsets of a knowledge base by taking the probabilities of the formulas into account accordingly. This gives rise to a probabilistic entailment relation that can be used for uncertain reasoning. Our approach is a generalisation of many concrete probabilistic enrichments of existing approaches, such as ProbLog (an approach to probabilistic logic programming) and the constellation approach to abstract argumentation. We analyse general properties of our approach and provide some insights into novel instantiations that have not been investigated yet.

1. Introduction

The ability to reason under uncertainty is a core requirement for most intelligent systems and many approaches for uncertain reasoning have been proposed in the area of *knowledge representation and reasoning* (KR) and *artificial intelligence* (AI) in general, see e. g. [43, 42, 3] for some textbooks on that topic. In general, we can distinguish between *qualitative* uncertain reasoning and *quantitative* uncertain reasoning. The former encompasses approaches such as *default logic* [51], *answer set programming* [24, 22], or *abstract argumentation* [16]. The latter makes use of formalisms such as *probability theory* [43, 42], *fuzzy logic* [26], or *Dempster-Shafer theory* [56]. A common approach to define a new quantitative model for uncertain reasoning is to take some non-quantitative approach—which may either be a qualitative model as mentioned before or something completely different such as propositional logic—add quantities to the syntax and define a new quantitative semantics on top of that. This approach is followed by e. g. probabilistic logics [41, 27]; distribution semantics for logic programming [55], then implemented in ProbLog [11]; P-log [2]; probabilistic approaches to formal argumentation [35, 29, 30, 53, 44], and many more.

In this paper we aim at unifying many of the aforementioned approaches and defining a general methodology for reasoning with quantitative uncertainty. This allows for a general study of its properties while abstracting away from any specific instantiation. We focus on probability theory as a means for quantitative uncertain reasoning but a similar methodology can be defined by building on other formalisms such as fuzzy logic or Dempster-Shafer theory. We start by considering an arbitrary *base logic* and define its *probabilistic augmentation* by extending the syntax to allow for annotated probabilities on each formula. Therefore, a knowledge base of probabilistic augmentation consists of a set of formulas, each annotated with a probability. We define a general probabilistic semantics on top of the built-in semantics of the base logic by (1) considering each subset of the knowledge base, (2) performing ordinary inference within the subset, and (3) accumulating the inferences by taking the probabilities into account. This gives us a general methodology for defining probabilistic versions of a wide spectrum of existing knowledge representation formalisms, and is inspired by many concrete realisations such as the distribution semantics for logic programming [55] and probabilistic data bases [57], cf. Section 6.

In order to illustrate our general methodology we provide instantiations of it using propositional logic, logic programming, and abstract argumentation. For those formalisms we provide the necessary preliminaries in Section 2. Afterwards, we present the contributions of this paper, namely:

1. We define the syntax and semantics of the *probabilistic augmentation* of a general knowledge representation formalism as a probabilistic version of it, and show how our general approach subsumes existing approaches to probabilistic reasoning (Section 3).
2. We provide an extensive analysis of our approach in terms of robustness of reasoning results (Section 4).
3. We discuss novel instantiations of our approach that have not been investigated yet (Section 5).

Section 6 reviews related works and Section 7 concludes this paper with a summary.

An extended abstract of this work has been published in [7] and a preliminary workshop version in [8].

2. Knowledge representation formalisms

We consider a very general definition for a logic. For a set S let 2^S denote its power set.

Definition 1. *A* logic $\mathcal{L}$ *is a tuple* $\mathcal{L} = (\mathcal{W}, \mathcal{V}, \models)$ *where* $\mathcal{W}$ *is the set of well-formed formulas,* $\mathcal{V}$ *is the set of "inferrable" formulas, and* $\models \subseteq 2^{\mathcal{W}} \times \mathcal{V}$ *is an inference relation.*

As we aim to model a wide range of logics we explicitly distinguish between well-formed formulas $\mathcal{W}$ and formulas that can be inferred in the formalism $\mathcal{V}$. For example, note that most approaches to logic programming (see Section 2.2) have rules as well-formed formulas, but inference is usually defined for ground atoms.

We write $\mathcal{K} \models \phi$ ("$\mathcal{K}$ entails ϕ") instead of $(\mathcal{K}, \phi) \in \models$ for $\mathcal{K} \subseteq \mathcal{W}$, $\phi \in \mathcal{V}$. $\mathcal{K} \subseteq \mathcal{W}$ is $\models$-inconsistent if $\mathcal{K} \models \phi$ for all $\phi \in \mathcal{V}$; otherwise $\mathcal{K}$ is $\models$-consistent.

Let us give instantiations of propositional logic, logic programming (Prolog and Answer Set Programming), and abstract argumentation in this simple framework.

2.1. Propositional logic

For a set of propositional atoms At we define propositional logic

$$\mathcal{L}_P(\mathsf{At}) = (\mathcal{W}_P(\mathsf{At}), \mathcal{V}_P(\mathsf{At}), \models_P)$$

as follows. Let $\mathcal{W}_P(\mathsf{At})$ be the propositional language generated using At and the usual connectives $\wedge$, $\vee$, $\Rightarrow$, and $\neg$. An interpretation ω is a function $\omega : \mathsf{At} \rightarrow \{\mathsf{true}, \mathsf{false}\}$. With slight abuse of notation we abbreviate (let $\phi, \psi \in \mathcal{W}_P$ and $\Psi \subseteq \mathcal{W}_P$)

$$\begin{aligned}
\omega(\psi \wedge \phi) &= \mathsf{true} \text{ iff } \omega(\psi) = \mathsf{true} \text{ and } \omega(\phi) = \mathsf{true} \\
\omega(\psi \vee \phi) &= \mathsf{true} \text{ iff } \omega(\psi) = \mathsf{true} \text{ or } \omega(\phi) = \mathsf{true} \\
\omega(\psi \Rightarrow \phi) &= \mathsf{true} \text{ iff } \omega(\psi) = \mathsf{false} \text{ or } \omega(\phi) = \mathsf{true} \\
\omega(\neg\phi) &= \mathsf{true} \text{ iff } \omega(\phi) = \mathsf{false} \\
\omega(\Psi) &= \mathsf{true} \text{ iff for all } \psi \in \Psi, \omega(\psi) = \mathsf{true}
\end{aligned}$$

We define $\mathcal{V}_P(\mathsf{At}) = \mathcal{W}_P(\mathsf{At})$ and classical entailment $\models_P$ via $\mathcal{K} \models_P \phi$ iff for all interpretations ω with $\omega(\mathcal{K}) = \mathsf{true}$, $\omega(\phi) = \mathsf{true}$ as well. With this notion, observe that $\mathcal{K} \subseteq \mathcal{W}_P(\mathsf{At})$ is $\models_P$-inconsistent iff it is classically inconsistent.

Example 1. *Let us consider the following example inspired by [49] and modelling a simple e-mail spam filter. Let $\mathcal{K}$ be the knowledge base consisting of the following formulas:*

$$\begin{aligned}&\mathsf{sc} \Rightarrow \mathsf{sp}\\&\mathsf{ss} \Rightarrow \mathsf{sp}\\&\mathsf{us} \Rightarrow \mathsf{sp} \vee \mathsf{jo}\\&\mathsf{sc}\\&\mathsf{ss}\end{aligned}$$

where sc *means "suspicious content",* ss *means "suspicious subject",* sp *means "spam",* us *means "unknown sender", and* jo *means "job offer". For example, the third formula can then be read as "a mail from an unknown sender is either spam or a job offer". From $\mathcal{K}$ we can infer* sp *($\mathcal{K} \models_P \mathsf{sp}$),* sc *($\mathcal{K} \models_P \mathsf{sc}$), and* ss *($\mathcal{K} \models_P \mathsf{ss}$).*

2.2. *Logic programming*

We now consider logic programming [34, 4, 24, 23, 22]. Let Pred be a set of predicate symbols, U a set of constant symbols, and V a set of variables. For every predicate symbol $p \in \mathsf{Pred}$ we denote by $\mathsf{arity}(p)$ its arity. An atom $p(t_1, \ldots, t_n)$ is a predicate symbol $p \in \mathsf{Pred}$ with $\mathsf{arity}(p) = n$ and $t_1, \ldots, t_n \in U \cup V$. A *literal* is either an atom $p(t_1, \ldots, t_n)$ or its classical negation $\neg p(t_1, \ldots, t_n)$. A *rule* r has the form

$$r : H \leftarrow A_1, \ldots, A_n, \mathsf{not}\ B_{n+1}, \ldots, \mathsf{not}\ B_m$$

with literals $H, A_1, \ldots, A_n, B_{n+1}, \ldots, B_m$. The literal $H = \mathit{head}(r)$ is called *head* of the rule and $\mathit{body}(r) = \{A_1, \ldots, A_n, \mathsf{not}\ B_{n+1}, \ldots, \mathsf{not}\ B_m\}$ is called *body* of the rule. We also differentiate between the positive body $\mathit{body}^+(r) = \{A_1, \ldots, A_n\}$ and the negative body $\mathit{body}^-(r) = \{\mathsf{not}\ B_{n+1}, \ldots, \mathsf{not}\ B_m\}$.

A rule is *safe* if every variable appearing in it, also appears in $\mathit{body}^+(r)$. A rule is *normal* if it does not contain classical negation $\neg$. If $m = n = 0$ we write H instead of $H \leftarrow$ and call this rule a *fact*. A rule is *ground* if it does not contain any variables. For a set P of rules we denote by $\mathit{ground}(P)$ its *grounding*, i. e., the set of all rules that can be obtained by some rule in P by substituting all variables homogeneously by constants.

We consider two concrete logics as instances of logic programming. First, we define[1] Prolog [34, 4] as the general logic

$$\mathcal{L}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V) = \\ (\mathcal{W}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V), \mathcal{V}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V), \models_{\mathsf{Prolog}})$$

where

- $\mathcal{W}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V)$ is the set of all safe and normal rules as defined above and
- $\mathcal{V}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V)$ is the set of all ground atoms.

For a set $P \subseteq \mathcal{W}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V)$ (also called Prolog program) the relation $\models_{\mathsf{Prolog}}$ is inductively defined via:[2]

1. for all $r \in P$ with $\mathit{body}(r) = \emptyset$, $P \models_{\mathsf{Prolog}} \mathit{head}(r)$.
2. $P \models_{\mathsf{Prolog}} H$ for $H \leftarrow A_1, \ldots, A_n, \mathsf{not}\ B_{n+1}, \ldots, \mathsf{not}\ B_m \in \mathit{ground}(P)$, if $P \models_{\mathsf{Prolog}} A_1, \ldots, P \models_{\mathsf{Prolog}} A_n$, $P \not\models_{\mathsf{Prolog}} B_{n+1}, \ldots, P \not\models_{\mathsf{Prolog}} B_m$.

If there is an H for which $P \models_{\mathsf{Prolog}} H$ cannot be decided "with a finite derivation" (such as in $P = \{H \leftarrow \mathsf{not}\ H.\}$, we define $P \models_{\mathsf{Prolog}} \phi$ for all $\phi \in \mathcal{W}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V)$ (P is $\models_{\mathsf{Prolog}}$-inconsistent).

Example 2 (Evolved from [11, Example 1])**.** *Let us consider the following knowledge base represented as a Prolog program $P \subseteq \mathcal{W}_{\mathit{Prolog}}(\mathit{Pred}, U, V)$:*

$$\begin{aligned}
&\mathit{likes}(X, Y) \leftarrow \mathit{friendof}(X, Y)\\
&\mathit{likes}(X, Y) \leftarrow \mathit{friendof}(X, Z), \mathit{likes}(Z, Y)\\
&\mathit{friendof}(john, mary)\\
&\mathit{friendof}(mary, pedro)
\end{aligned}$$

Consistently with Prolog semantics, $P \models_{\mathit{Prolog}} \mathit{likes}(john, mary)$, $P \models_{\mathit{Prolog}} \mathit{likes}(mary, pedro)$, $P \models_{\mathit{Prolog}} \mathit{likes}(john, pedro)$.

[1] We simplify the syntax for matters of presentation.

[2] Note that we define the operational semantics of Prolog in an informal manner sufficient for this paper, for details please see [4].

Secondly, we consider answer set programming [24, 23, 22] and define

$$\mathcal{L}_{\mathsf{ASP}}(\mathsf{Pred}, U, V) = \\ (\mathcal{W}_{\mathsf{ASP}}(\mathsf{Pred}, U, V), \mathcal{V}_{\mathsf{ASP}}(\mathsf{Pred}, U, V), \models_{\mathsf{ASP}})$$

where

- $\mathcal{W}_{\mathsf{ASP}}(\mathsf{Pred}, U, V)$ is the set of all safe rules and
- $\mathcal{V}_{\mathsf{ASP}}(\mathsf{Pred}, U, V)$ is the set of all ground literals.

For a set $P \subseteq \mathcal{W}_{\mathsf{ASP}}(\mathsf{Pred}, U, V)$ (also called answer set program) and a set M of ground literals, the *reduct* P^M is defined via

$$P^M = \{\mathit{head}(r) \leftarrow \mathit{body}^+(r) \mid r \in \mathit{ground}(P), \\ M \cap \mathit{body}^-(r) = \emptyset\}$$

A set M of ground literals is called *answer set* if it is the minimal (wrt. set inclusion) model of P^M. Then $P \models_{\mathsf{ASP}} H$ for a ground literal H iff $H \in M$ for all answer sets M.[3] If there are no answer sets in P, we define $P \models_{\mathsf{ASP}} \phi$ for all $\phi \in \mathcal{V}_{\mathsf{ASP}}(\mathsf{Pred}, U, V)$ (P is $\models_{\mathsf{ASP}}$-inconsistent).

Example 3. *Let us consider the following answer set program $\mathcal{K}$:*

$$\begin{aligned}&\mathit{drill} \leftarrow \mathit{alarm}, \mathit{not\ real}\\ &\mathit{real} \leftarrow \mathit{alarm}, \mathit{not\ drill}\\ &\mathit{alarm}\end{aligned}$$

Remembering that $\models_{\mathsf{ASP}}$ is a skeptical relation, we have only $\mathcal{K} \models_{\mathsf{ASP}} \mathit{alarm}$ as there are the following two answer sets: $\{\mathit{alarm}, \mathit{real}\}$ and $\{\mathit{alarm}, \mathit{drill}\}$.

2.3. Abstract argumentation

We represent abstract argumentation frameworks [16] as a logic as follows. Let $\mathcal{A}$ be a set of arguments (abstract atomic entities) and define the language

$$\mathcal{W}_{\mathsf{AAF}}(\mathcal{A}) = \mathcal{A} \cup (\mathcal{A} \times \mathcal{A})$$

[3] Observe that we define $\models_{\mathsf{ASP}}$ to be a skeptical inference relation, but using a credulous approach can be defined analogously by requiring that H is contained in *some* answer set.

In other words, $\mathcal{W}_{\mathsf{AAF}}(\mathcal{A})$ is the set of all arguments and all pairs of arguments. Each $\mathcal{K} \subseteq \mathcal{W}_{\mathsf{AAF}}(\mathcal{A})$ defines a directed graph $G_{\mathcal{K}} = (V_{\mathcal{K}}, E_{\mathcal{K}})$ via

$$\begin{aligned} V_{\mathcal{K}} &= \mathcal{K} \cap \mathcal{A} \\ E_{\mathcal{K}} &= \mathcal{K} \cap (V_{\mathcal{K}} \times V_{\mathcal{K}}) \end{aligned}$$

A set $M \subseteq V_{\mathcal{K}}$ (also called *extension*) is *conflict-free* if for all $\mathbf{a}, \mathbf{b} \in M$, $(\mathbf{a}, \mathbf{b}) \notin E_{\mathcal{K}}$. An argument $\mathbf{a} \in \mathcal{A}$ is *acceptable* wrt. $M \subseteq V_{\mathcal{K}}$ if for all $(\mathbf{b}, \mathbf{a}) \in E_{\mathcal{K}}$ there is $\mathbf{c} \in M$ with $(\mathbf{c}, \mathbf{b}) \in E_{\mathcal{K}}$. The extension M is admissible if every $\mathbf{a} \in M$ is acceptable wrt. M. The four major semantics of abstract argumentation frameworks are the complete, grounded, preferred, and stable semantics defined as follows. For $M \subseteq V_{\mathcal{K}}$ we say that

1. M is a *complete* extension if it is admissible and and every $\mathbf{a} \in \mathcal{A}$ that is acceptable wrt. M belongs to M;
2. M is the *grounded* (GR) extension if it is complete and minimal wrt. set inclusion;
3. M is a *preferred* (PR) extension if it is complete and maximal wrt. set inclusion;
4. M is a *stable* (ST) extension if it is conflict-free and for all $\mathbf{a} \in V_{\mathcal{K}} \setminus M$, there is $\mathbf{b} \in M$ with $(\mathbf{b}, \mathbf{a}) \in E_{\mathcal{K}}$.

Note that the grounded extension M is uniquely determined [16]. The set $\mathcal{V}_{\mathsf{AAF}}(\mathcal{A})$ is simply $\mathcal{A}$.

Let $\sigma \in \{\mathsf{GR}, \mathsf{PR}, \mathsf{ST}\}$ be any of the above semantical notions. We define the inference relation $\models^{\sigma}_{\mathsf{AAF}}$ via $\mathcal{K} \models^{\sigma}_{\mathsf{AAF}} \mathbf{a}$ if for all σ-extensions M of $G_{\mathcal{K}}$, $\mathbf{a} \in M$. Note that with this definition, an abstract argumentation framework $\mathcal{K}$ is $\models^{\sigma}_{\mathsf{AAF}}$-inconsistent iff there is no σ-extension (this can actually only happen for stable semantics). The above definitions give us for any semantics σ a logic

$$\mathcal{L}^{\sigma}_{\mathsf{AAF}}(\mathcal{A}) = (\mathcal{W}_{\mathsf{AAF}}(\mathcal{A}), \mathcal{V}_{\mathsf{AAF}}(\mathcal{A}), \models^{\sigma}_{\mathsf{AAF}}).$$

Example 4. *Consider the argumentation framework in Figure 1 formed of three arguments:* ***a*** *stating that tomorrow will not rain;* ***b*** *stating that tomorrow will rain; and* **c** *stating that tomorrow we should have a barbecue.* ***a*** *and* ***b*** *are mutually exclusive,*[4] *and* ***b*** *provides enough reasons for not having a barbecue tomorrow.*

[4] We consider the entire day as a single atomic unit of time.

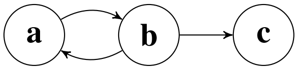

Figure 1: An argumentation framework

This setting can be represented in $\mathcal{L}^{\sigma}_{\mathsf{AAF}}(\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\})$ *as the knowledge base* $\mathcal{K}$ *via*

$$\begin{aligned}\mathcal{K} = \{&\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\\ &(\boldsymbol{a},\boldsymbol{b}),(\boldsymbol{b},\boldsymbol{a}),(\boldsymbol{b},\boldsymbol{c})\}\end{aligned}$$

The grounded extension here is empty, and the following are both preferred and stable extensions: $\{\boldsymbol{a},\boldsymbol{c}\}$*, and* $\{\boldsymbol{b}\}$*. It is therefore immediate to see that for any* $\sigma \in \{\mathsf{GR},\mathsf{PR},\mathsf{ST}\}$*, the set of inferences using* $\models^{\sigma}_{\mathsf{AAF}}$ *is empty.*

3. A general probabilistic approach

Let $\mathcal{L} = (\mathcal{W},\mathcal{V},\models)$ be some logic, which will also be referred to as *base logic* in the following. We define its *probabilistic augmentation* $\mathcal{Z}(\mathcal{L}) = (\widehat{\mathcal{W}},\widehat{\mathcal{V}},\widehat{\models})$ as follows.

The languages $\widehat{\mathcal{W}}$ and $\widehat{\mathcal{V}}$ consist of the quantification of formulas of $\mathcal{L}$ with probabilities:

$$\begin{aligned}\widehat{\mathcal{W}} &= \{\phi : p \mid \phi \in \mathcal{W}, p \in [0,1]\}\\ \widehat{\mathcal{V}} &= \{\phi : p \mid \phi \in \mathcal{V}, p \in [0,1]\}\end{aligned}$$

The semantics of $\mathcal{Z}(\mathcal{L})$ are defined in terms of probabilities of subsets of a knowledge base $\mathcal{K} \subseteq \widehat{\mathcal{W}}$. For every $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ define

$$\mathcal{K}\downarrow = \{\phi \mid \phi : p \in \mathcal{K}\} \subseteq \mathcal{W} \tag{1}$$

In other words, $\mathcal{K}\downarrow$ is the *flattened*—i.e. without probabilities—version of the knowledge base $\mathcal{K}$. We define now the *general probability* $P_{\mathcal{K}}$ of subsets of a probabilistic knowledge base $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ via

$$P_{\mathcal{K}}(\mathcal{K}') = \prod_{\phi\,:\,p\in\mathcal{K}'} p \prod_{\phi\,:\,p\in\mathcal{K}\setminus\mathcal{K}'} (1-p) \tag{2}$$

for all $\mathcal{K}' \subseteq \mathcal{K}$. Observe that $P_{\mathcal{K}}$ is indeed a probability distribution over subsets of $\mathcal{K}$.

Proposition 1. *For every $\mathcal{K} \subseteq \widehat{\mathcal{W}}$, $\sum_{\mathcal{K}' \subseteq \mathcal{K}} P_{\mathcal{K}}(\mathcal{K}') = 1$.*

Proof. We prove the statement through induction by $|\mathcal{K}| = n$.

- $n = 1$. Assume $\mathcal{K} = \{\phi : p\}$ and consider

$$\begin{aligned}\sum_{\mathcal{K}' \subseteq \mathcal{K}} P_{\mathcal{K}}(\mathcal{K}') &= \sum_{\mathcal{K}' \subseteq \mathcal{K}} \left(\prod_{\phi\,:\,p \in \mathcal{K}'} p \prod_{\phi\,:\,p \in \mathcal{K} \setminus \mathcal{K}'} (1-p) \right) \\ &= \left(\prod_{\phi\,:\,p \in \emptyset} p \prod_{\phi\,:\,p \in \mathcal{K}} (1-p) \right) \\ &\quad + \left(\prod_{\phi\,:\,p \in \mathcal{K}} p \prod_{\phi\,:\,p \in \emptyset} (1-p) \right) \\ &= (1-p) + p = 1\end{aligned}$$

- $n \rightarrow n+1$. Assume $\mathcal{K} = \tilde{\mathcal{K}} \cup \{\psi\}$ with $\psi = \phi : p$, $|\tilde{\mathcal{K}}| = n$ and consider

$$\begin{aligned}\sum_{\mathcal{K}' \subseteq \mathcal{K}} P_{\mathcal{K}}(\mathcal{K}') &= \sum_{\mathcal{K}' \subseteq \mathcal{K}, \psi \in \mathcal{K}'} P_{\mathcal{K}}(\mathcal{K}') + \sum_{\mathcal{K}' \subseteq \mathcal{K}, \psi \notin \mathcal{K}'} P_{\mathcal{K}}(\mathcal{K}') \\ &= \sum_{\mathcal{K}' \subseteq \tilde{\mathcal{K}}} p P_{\tilde{\mathcal{K}}}(\mathcal{K}') + \sum_{\mathcal{K}' \subseteq \tilde{\mathcal{K}}} (1-p) P_{\tilde{\mathcal{K}}}(\mathcal{K}') \\ &= p \underbrace{\sum_{\mathcal{K}' \subseteq \tilde{\mathcal{K}}} P_{\tilde{\mathcal{K}}}(\mathcal{K}')}_{=1} + (1-p) \underbrace{\sum_{\mathcal{K}' \subseteq \tilde{\mathcal{K}}} P_{\tilde{\mathcal{K}}}(\mathcal{K}')}_{=1} \\ &= p + (1-p) = 1\end{aligned}$$

□

Based on the general probability $P_{\mathcal{K}}$ we can define the *degree of belief* of any formula $\phi \in \mathcal{V}$ wrt. $\mathcal{K}$ via

$$\Pi_{\mathcal{K}}(\phi) = \sum_{\mathcal{K}' \subseteq \mathcal{K}, \mathcal{K}' \downarrow \models \phi} P_{\mathcal{K}}(\mathcal{K}')$$

In other words, a probabilistic knowledge base $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ defines a probability distribution over all subsets of $\mathcal{K}$. For each subset $\mathcal{K}' \subseteq \mathcal{K}$, we consider its flattened version $\mathcal{K}' \downarrow$ and decide using the base logic $\mathcal{L}$, whether $\mathcal{K}' \downarrow$ infers ϕ. We sum up the probabilities of all subsets where this is the case in order to obtain the degree of belief of ϕ wrt. the probabilistic knowledge base $\mathcal{K}$.

Based on $\Pi_{\mathcal{K}}$ we define probabilistic inference $\widehat{\models}$ via

$$\mathcal{K} \mathrel{\widehat{\models}} \phi \,:\, p \text{ if } \Pi_{\mathcal{K}}(\phi) = p$$

for all $\mathcal{K} \subseteq \widehat{\mathcal{W}}$.

Before we continue with some concrete examples of probabilistic augmentations, we make some first general analysis. Note that we defined $\Pi_{\mathcal{K}}$ as a degree of belief and not as a probability. This is sensible as we did not require our base logic to possess some form of negation. However, for any reasonable definition of probability, we need some form of complement operator.

Definition 2. *Let $\mathcal{L} = (\mathcal{W}, \mathcal{V}, \models)$ be a logic.*

1. *We say that $\mathcal{L}$ has a* weak negation operator $-$ *if, for every $\models$-consistent $\mathcal{K} \subseteq \mathcal{W}$ and $\phi \in \mathcal{V}$ such that $-\phi \in \mathcal{V}$ as well, it is not the case that both $\mathcal{K} \models \phi$ and $\mathcal{K} \models -\phi$.*
2. *We say that $\mathcal{L}$ has a* strict negation operator $-$ *if, for every $\models$-consistent $\mathcal{K} \subseteq \mathcal{W}$ and $\phi \in \mathcal{V}$ such that $-\phi \in \mathcal{V}$ as well, either $\mathcal{K} \models \phi$ or $\mathcal{K} \models -\phi$ (but not both).*
3. *We say that $\mathcal{L}$ has a* universal negation $-$ *if, for all $\phi \in \mathcal{V}$, $-\phi \in \mathcal{V}$.*

Every strict negation is also weak and a logic may possess multiple types of negations.

Example 5. *We revisit our example logics from Section 2.*

1. *A propositional logic $\mathcal{L}_P(\mathsf{At}) = (\mathcal{W}_P(\mathsf{At}), \mathcal{V}_P(\mathsf{At}), \models_P)$ (under classical semantics) has a weak negation $\neg$ as it can be the case that $\mathcal{K} \not\models_P \phi$ and $\mathcal{K} \not\models_P \neg\phi$ for some formula ϕ. Furthermore, $\neg$ is a universal negation.*
2. *Prolog's negation* not *is trivially strict (and weak) as for every $\phi \in \mathcal{V}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V)$ it is not the case that* not *$\phi \in \mathcal{V}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V)$ (negation-as-failure is only used in rule bodies). It is obviously not universal.*
3. *Answer set programming has two negations $\neg$ and* not*. The former is weak and universal, the latter is (trivially) strict and not universal.*
4. *Abstract argumentation has no negation operator.*

Having a certain type of negation constrains the degrees of belief in complementary statements as follows.

Proposition 2. *Let $\mathcal{L}$ be a logic with a negation operator $-$.*

1. *If $-$ is strict then $\Pi_{\mathcal{K}}(\phi) + \Pi_{\mathcal{K}}(-\phi) = 1$ for every $\models$-consistent $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ and $\phi \in \mathcal{V}$ such that $-\phi \in \mathcal{V}$ as well.*
2. *If $-$ is weak then $\Pi_{\mathcal{K}}(\phi) + \Pi_{\mathcal{K}}(-\phi) \leq 1$ for every $\models$-consistent $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ and $\phi \in \mathcal{V}$ such that $-\phi \in \mathcal{V}$ as well.*

Proof. Let us consider 1.

$$\begin{aligned}
&\Pi_{\mathcal{K}}(\phi) + \Pi_{\mathcal{K}}(-\phi) = \\
=&\sum_{\mathcal{K}' \subseteq \mathcal{K}, \mathcal{K}' \downarrow \models \phi} P_{\mathcal{K}}(\mathcal{K}') + \sum_{\mathcal{K}' \subseteq \mathcal{K}, \mathcal{K}' \downarrow \models -\phi} P_{\mathcal{K}}(\mathcal{K}') = \\
&\textit{(since $-$ is strict)} \\
=&\sum_{\mathcal{K}' \subseteq \mathcal{K}} P_{\mathcal{K}}(\mathcal{K}') = 1 \ \textit{(from Proposition 1)}
\end{aligned}$$

Let us consider 2.

$$\begin{aligned}
&1 = \\
=&\sum_{\mathcal{K}' \subseteq \mathcal{K}} P_{\mathcal{K}}(\mathcal{K}') \ \textit{(from Proposition 1)} \\
=&\sum_{\mathcal{K}' \subseteq \mathcal{K}, \mathcal{K}' \downarrow \models \phi} P_{\mathcal{K}}(\mathcal{K}') + \sum_{\mathcal{K}' \subseteq \mathcal{K}, \mathcal{K}' \downarrow \models -\phi} P_{\mathcal{K}}(\mathcal{K}') + \\
&+ \sum_{\mathcal{K}' \subseteq \mathcal{K}, \mathcal{K}' \downarrow \not\models \phi \vee -\phi} P_{\mathcal{K}}(\mathcal{K}') = \\
=&\Pi_{\mathcal{K}}(\phi) + \Pi_{\mathcal{K}}(-\phi) + \sum_{\mathcal{K}' \subseteq \mathcal{K}, \mathcal{K}' \downarrow \not\models \phi \vee -\phi} P_{\mathcal{K}}(\mathcal{K}')
\end{aligned}$$

hence $\Pi_{\mathcal{K}}(\phi) + \Pi_{\mathcal{K}}(-\phi) \leq 1$. □

Among others, our approach generalises ProbLog and the constellation approach to abstract argumentation.

3.1. ProbLog

ProbLog [11] (*Probabilistic Prolog*) is a probabilistic logic programming language building on Prolog and Sato's distribution semantics [55], in which facts can be annotated with the probability that they hold. Although ProbLog is not the only implementation of distribution semantics, e.g. [21, 9], for the purpose of this paper we will focus on it as an example.

In our general framework it can be easily defined as follows.

Definition 3. *Let* $\textsf{\textit{Pred}}$ *be a set of predicate symbols,* U *a set of constant symbols, and* V *a set of variables. The logic* $\mathcal{Z}(\mathcal{L}_{\textsf{\textit{Prolog}}}(\textsf{\textit{Pred}}, U, V))$ *is called* ProbLog.[5]

In other words, in ProbLog a knowledge base consists of a set of safe and normal rules, each annotated with a probability.

Let us redefine the Prolog program introduced in Example 2 as a knowledge base and its probabilistic augmentation.

Example 6 (Evolved from [11, Example 1])**.** *Let us extend Example 2 with probabilities, and let*

$$\begin{aligned}\mathcal{K} \subseteq & \mathcal{Z}(\mathcal{L}_{\textsf{\textit{Prolog}}}(\textsf{\textit{Pred}}, U, V)) \\ & = \mathcal{Z}((\mathcal{W}_{\textsf{\textit{Prolog}}}(\textsf{\textit{Pred}}, U, V), \mathcal{V}_{\textsf{\textit{Prolog}}}(\textsf{\textit{Pred}}, U, V), \models_{\textsf{\textit{Prolog}}}))\end{aligned}$$

be the knowledge base consisting of the following formulas:

$$\begin{array}{ll} \textsf{\textit{likes}}(X, Y) \leftarrow \textsf{\textit{friendof}}(X, Y) & : 1.0 \\ \textsf{\textit{likes}}(X, Y) \leftarrow \textsf{\textit{friendof}}(X, Z), \textsf{\textit{likes}}(Z, Y) & : 0.8 \\ \textsf{\textit{friendof}}(john, mary) & : 0.5 \\ \textsf{\textit{friendof}}(mary, pedro) & : 0.5 \end{array}$$

[5] As discussed in `https://dtai.cs.kuleuven.be/problog/tutorial/advanced/04_prolog.html` (on 7 May 2018), in the ProbLog2 python interface and implementation available at `https://dtai.cs.kuleuven.be/problog/index.html#download` (on 7 May 2018)—and differently from other ProbLog implementations that use Prolog based interface—each grounding of a query is restricted to occur only once as a result, even if their are multiple proofs. Moreover, ProbLog extends Prolog's tabling to support cyclic ProbLog programs [39]. We will not discuss further those aspects as they are just implementation details.

Consistently with ProbLog semantics,[6] $\mathcal{K} \mathrel{\widehat{\models_{\mathit{Prolog}}}} \mathit{likes}(john, pedro) : 0.2$*. Indeed, the two* $\mathcal{K}' \subseteq \mathcal{K}$ *such that* $\mathcal{K}' \downarrow\ \models_{\mathit{Prolog}} \mathit{likes}(john, pedro)$ *are (we abbreviate all predicate and atoms by their first letter):*

$$\mathcal{K}'_1 = \left\{ \begin{array}{rl} \mathit{l}(j,p) \leftarrow \mathit{f}(j,m), \mathit{l}(m,p) & : 0.8 \\ \mathit{f}(j,m) & : 0.5 \\ \mathit{l}(m,p) \leftarrow \mathit{f}(m,p) & : 1.0 \\ \mathit{f}(m,p) & : 0.5 \end{array} \right\}$$

and $\mathcal{K}'_2 = \mathcal{K}'_1 \cup \{\mathit{l}(j,m) \leftarrow \mathit{f}(j,m)\ : 1.0\}$*.*

3.2. The constellation approach to abstract argumentation

The constellation approach to abstract argumentation [35, 29, 30] is an extension of abstract argumentation that adds probabilities to arguments and attacks. In our general framework it can be defined as follows.

Definition 4. *Let σ be some semantics for abstract argumentation. The logic $\mathcal{Z}(\mathcal{L}^{\sigma}_{\mathit{AAF}})$ is called the logic of* probabilistic argumentation frameworks.

Example 7. *Consider the probabilistic argumentation framework in Figure 2 that extends Example 4. Informally, this representation means that argument $\boldsymbol{a}$ is "present" with probability* 0.9*, argument $\boldsymbol{b}$ is "present" with probability* 0.6 *and the attack $(\boldsymbol{a}, \boldsymbol{b})$ is "present" with* conditional *probability* 0.3*, given that both $\boldsymbol{a}$ and $\boldsymbol{b}$ are present. This setting can be represented in $\mathcal{Z}(\mathcal{L}^{\sigma}_{\mathit{AAF}})(\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\})$ as the knowledge base $\mathcal{K}$ consisting of the formulas*

$$\begin{array}{ll} \boldsymbol{a} & : 0.9 \\ \boldsymbol{b} & : 0.6 \\ \boldsymbol{c} & : 0.8 \\ (\boldsymbol{a}, \boldsymbol{b}) & : 0.3 \\ (\boldsymbol{b}, \boldsymbol{a}) & : 0.6 \\ (\boldsymbol{b}, \boldsymbol{c}) & : 0.7 \end{array}$$

The way we defined the induced graph $G_{\mathcal{K}}$ (see Section 2.3) ensures that probabilities of attacks are indeed interpreted as conditional *probabilities.*

[6] `https://dtai.cs.kuleuven.be/problog/editor.html#task=prob&hash=1cee65d0c7262ad6ba48092942c46088` (on 7 May 2018)

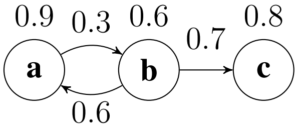

Figure 2: A simple argumentation framework

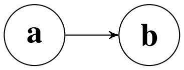

Figure 3: An argumentation framework

Note that the above definition captures the original definition of [35] where probabilities of arguments are independent of each other and the probabilities of attacks are independent given that both arguments of an attack are present. With a more general base logic we are also able to capture the more general setting of [30] where probabilistic dependencies between components are allowed. For that let $\mathcal{A}$ again be a set of arguments and define the language $\mathcal{W}^{\mathsf{ind}}_{\mathsf{AAF}}(\mathcal{A})$ to be the set of normal logic programming rules with atoms $\mathcal{A}\cup(\mathcal{A}\times\mathcal{A})$. Therefore, $\mathcal{W}^{\mathsf{ind}}_{\mathsf{AAF}}(\mathcal{A})$ contains rules such as $(\mathbf{b},\mathbf{c}) \leftarrow \mathbf{a},\mathbf{d}$ meaning "if arguments **a** and **d** are present then the attack $(\mathbf{b},\mathbf{c})$ is present". Then a set $\mathcal{K} \subseteq \mathcal{W}^{\mathsf{ind}}_{\mathsf{AAF}}(\mathcal{A})$ defines an abstract argumentation framework $G_{\mathcal{K}} = (V_{\mathcal{K}}, E_{\mathcal{K}})$ via

$$\begin{aligned}V_{\mathcal{K}} &= \{\mathbf{a} \in \mathcal{A} \mid \mathcal{K} \models_{\mathsf{Prolog}} \mathbf{a}\}\\ E_{\mathcal{K}} &= \{(\mathbf{a},\mathbf{b}) \in (V_{\mathcal{K}} \times V_{\mathcal{K}}) \mid \mathcal{K} \models_{\mathsf{Prolog}} (\mathbf{a},\mathbf{b})\}\end{aligned}$$

Note that in the definition of $E_{\mathcal{K}}$ above we ensure that $G_{\mathcal{K}}$ is a well-defined graph by considering only $(\mathbf{a},\mathbf{b}) \in (V_{\mathcal{K}}\times V_{\mathcal{K}})$. Then $\mathcal{L}^{\sigma,\mathsf{ind}}_{\mathsf{AAF}}(\mathcal{A}) = (\mathcal{W}^{\mathsf{ind}}_{\mathsf{AAF}}(\mathcal{A}), \mathcal{V}_{\mathsf{AAF}}(\mathcal{A}), \models^{\sigma}_{\mathsf{AAF}}$) is defined analogously as $\mathcal{L}^{\sigma}_{\mathsf{AAF}}(\mathcal{A})$.

Example 8. *Consider the following set of rules from $\mathcal{K} \subseteq \mathcal{W}^{ind}_{AAF}(\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\})$:*

$$\begin{aligned}&(\boldsymbol{a},\boldsymbol{b}) \leftarrow \mathit{not}\,\boldsymbol{c}\\ &\boldsymbol{b} \leftarrow \boldsymbol{a}\\ &\boldsymbol{a}\end{aligned}$$

We obtain $G_{\mathcal{K}}$ as depicted in Figure 3.

We now consider the probabilistic augmentation $\mathcal{Z}(\mathcal{L}^{\sigma,\mathsf{ind}}_{\mathsf{AAF}})$ of $\mathcal{L}^{\sigma,\mathsf{ind}}_{\mathsf{AAF}}$ and show that it fully captures the framework of [30]. For that we generalise the definition of $P_{\mathcal{K}}$—see Equation (2)—for graphs $G = (V, E)$ via

$$P_{\mathcal{K}}(G) = \sum_{\mathcal{K}' \subseteq \mathcal{K}, G_{\mathcal{K}'} = G} P_{\mathcal{K}}(\mathcal{K}')$$

for all $\mathcal{K} \subseteq \mathcal{W}^{\mathsf{ind}}_{\mathsf{AAF}}(\mathcal{A})$. In other words, $P_{\mathcal{K}}(G)$ is the sum of the probabilities of all sets $\mathcal{K}' \subseteq \mathcal{K}$ that define the graph G.

Now recall that a *probabilistic attack graph* [30] is a tuple (V, E, Q) where (V, E) is a directed graph and Q is a probability function on subgraphs of (V, E). For a probabilistic attack graph (V, E, Q) let

$$sub((V, E)) = \{(V', E') \mid V' \subseteq V, E' \subseteq E \cap (V' \times V')\}$$

be the set of subgraphs of $G = (V, E)$. Then $Q : sub(G) \to [0, 1]$ with

$$\sum_{G' \in G} Q(G') = 1$$

The following theorem states that every probability function on subgraphs can be represented using the logic $\mathcal{Z}(\mathcal{L}^{\sigma,\mathsf{ind}}_{\mathsf{AAF}})$, in particular those that contain probabilistic dependencies between the components of the graph.

Theorem 1. *For every probabilistic attack graph* (V, E, Q) *we can define* $\mathcal{K} \subseteq \mathcal{W}^{\mathit{ind}}_{\mathit{AAF}}(V)$ *such that for every* $(V', E') \in sub((V, E))$

$$P_{\mathcal{K}}((V', E')) = Q((V', E')) \tag{3}$$

Proof. For $C \subseteq \{c, \mathsf{not}\ c \mid c \in V \cup E\}$ let

$$Q(X) = \sum_{(V', E') \in sub(G), \forall c \in X : c \in V' \cup E', \forall \mathsf{not}\ c \in X : c \notin V' \cup E'} Q(G')$$

be the probability of the set of components (vertices and/or edges) X. Let $C = \langle c_1, \ldots, c_n \rangle$ be an arbitrary ordering of the elements in $V \cup E$. Define a sequence $\mathcal{K}_i \in \mathcal{W}^{\mathsf{ind}}_{\mathsf{AAF}}(V)$ via

$$\mathcal{K}_0 = \{c_1 : Q(\{c_1\})\}$$

and for every $1 < i \leq n$

$$\mathcal{K}_i = \mathcal{K}_{i-1} \cup \{c_i \leftarrow \dot{c}_1, \ldots, \dot{c}_{i-1} : Q(c_i, \dot{c}_1, \ldots, \dot{c}_{i-1})/Q(\dot{c}_1, \ldots, \dot{c}_{i-1}) \mid \dot{c}_i \in \{c_i, \mathsf{not}\ c_i\}\}$$

In order to illustrate the above construction consider the abstract argumentation framework in Figure 3 and assume the order $C = \langle \mathbf{a}, (\mathbf{a}, \mathbf{b}), \mathbf{b} \rangle$. Then we get

$$\begin{aligned}
&\mathcal{K}_0 = \{\mathbf{a} : Q(\{\mathbf{a}\})\\
&\mathcal{K}_1 = \mathcal{K}_0 \cup \{\\
&\qquad (\mathbf{a}, \mathbf{b}) \leftarrow \mathbf{a} : Q(\{(\mathbf{a}, \mathbf{b}), \mathbf{a}\})/Q(\{\mathbf{a}\})\\
&\qquad (\mathbf{a}, \mathbf{b}) \leftarrow \mathsf{not}\ \mathbf{a} : Q(\{(\mathbf{a}, \mathbf{b}), \mathsf{not}\ \mathbf{a}\})/Q(\{\mathsf{not}\ \mathbf{a}\})\\
&\quad \}\\
&\mathcal{K}_2 = \mathcal{K}_1 \cup \{\\
&\qquad \mathbf{b} \leftarrow \mathbf{a}, (\mathbf{a}, \mathbf{b}) : Q(\{\mathbf{b}, (\mathbf{a}, \mathbf{b}), \mathbf{a}\})/Q(\{(\mathbf{a}, \mathbf{b}), \mathbf{a}\})\\
&\qquad \mathbf{b} \leftarrow \mathbf{a}, \mathsf{not}\ (\mathbf{a}, \mathbf{b}) : Q(\{\mathbf{b}, \mathsf{not}\ (\mathbf{a}, \mathbf{b}), \mathbf{a}\})/Q(\{\mathsf{not}\ (\mathbf{a}, \mathbf{b}), \mathbf{a}\})\\
&\qquad \mathbf{b} \leftarrow \mathsf{not}\ \mathbf{a}, (\mathbf{a}, \mathbf{b}) : Q(\{\mathbf{b}, (\mathbf{a}, \mathbf{b}), \mathsf{not}\ \mathbf{a}\})/Q(\{(\mathbf{a}, \mathbf{b}), \mathsf{not}\ \mathbf{a}\})\\
&\qquad \mathbf{b} \leftarrow \mathsf{not}\ \mathbf{a}, \mathsf{not}\ (\mathbf{a}, \mathbf{b}) : Q(\{\mathbf{b}, \mathsf{not}\ (\mathbf{a}, \mathbf{b}), \mathsf{not}\ \mathbf{a}\})/Q(\{\mathsf{not}\ (\mathbf{a}, \mathbf{b}), \mathsf{not}\ \mathbf{a}\})\\
&\quad \}
\end{aligned}$$

Note that $\mathcal{K}_n$ models a kind of binary decision tree over all subsets of the components of the graph (V, E). For the example above, Figure 4 shows an illustration of this binary decision tree. Each path from the root to the leafs corresponds to the construction of a subset of the components of the graph (V, E), where traversing left from a node means adding this component and traversing right means omitting this component. Each non-leaf node in this tree corresponds to a rule in $\mathcal{K}_n$, namely that one with the node label as head and the current path as premise (interpreting the omission of a component with not). The leaf nodes correspond to the final subsets.

Note that for every subgraph $G' = (V', E')$ there is a uniquely determined path in the decision tree. Furthermore, the probability of the corresponding set $V' \cup E'$ can be determined by multiplying the corresponding probabilities of the rules included in that path whenever we turn left and the complements of the probabilities of rules whenever we turn right. For example, for the set $\{\mathbf{a}, \mathbf{b}\}$ we

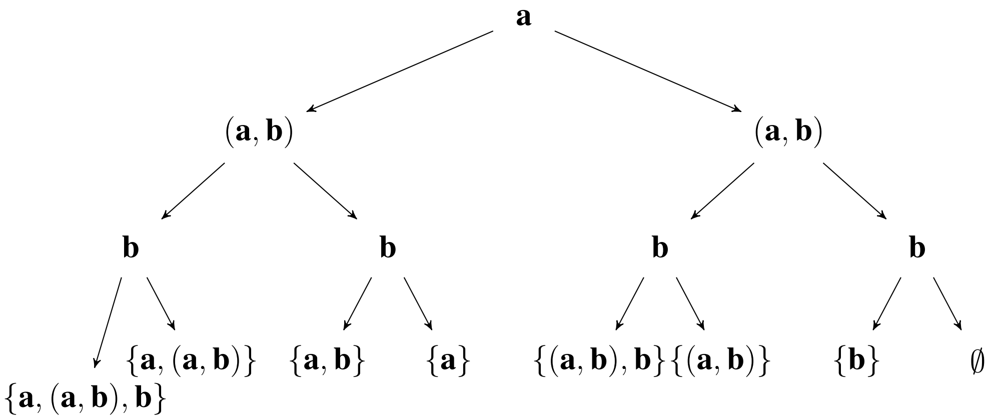

Figure 4: A binary decision tree for subset construction.

get

$$
\begin{aligned}
P_{\mathcal{K}_2}(\{\mathbf{a},\mathbf{b}\}) &= Q(\{\mathbf{a}\})(1-\frac{Q(\{(\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathbf{a}\})})\frac{Q(\{\mathbf{b},\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}\\
&= Q(\{\mathbf{a}\})(\frac{Q(\{\mathbf{a}\})}{Q(\{\mathbf{a}\})}-\frac{Q(\{(\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathbf{a}\})})\frac{Q(\{\mathbf{b},\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}\\
&= Q(\{\mathbf{a}\})\frac{Q(\{\mathbf{a}\})-Q(\{(\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathbf{a}\})}\frac{Q(\{\mathbf{b},\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}\\
&= Q(\{\mathbf{a}\})\frac{Q(\{\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathbf{a}\})}\frac{Q(\{\mathbf{b},\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}{Q(\{\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})}\\
&= Q(\{\mathbf{a},\mathsf{not}\ (\mathbf{a},\mathbf{b}),\mathbf{a}\})\\
&= Q((V',E'))
\end{aligned}
$$

Note that the probabilities of rules not on the path to a leaf do not influence the final probability (both subsets containing such rules and not containing such rules have no effect on the derived graph of the subset). Therefore $P_{\mathcal{K}_n}(V' \cup E') = P_{\mathcal{K}_n}((V',E'))$ and this naturally extends to the general case. □

4. Analysis

In the following, we analyse our general approach to probabilistic reasoning and investigate its properties. Note that for this investigation we assume any arbitrary base logic $\mathcal{L} = (\mathcal{W}, \mathcal{V}, \models)$, so our results are valid for a wide range of concrete knowledge representation formalisms.

Our first observation is that a probabilistic augmentation trivialises to the base logic if only the probability value 1 is used. More formally, we define the *trivial lifting* of $\mathcal{K} \subseteq \mathcal{W}$ as the knowledge base $\mathcal{K}^t = \{\phi : 1 \mid \phi \in \mathcal{K}\}$. In other words, $\mathcal{K}^t$ consists of all formulas of $\mathcal{K}$ with probability 1.

Theorem 2 (Trivialisation)**.** *Let $\mathcal{L} = (\mathcal{W}, \mathcal{V}, \models)$ be some logic and $\mathcal{K} \subseteq \mathcal{W}$. Then $\mathcal{K} \models \phi$ iff $\mathcal{K}^t \widehat{\models} \phi : 1$.*

Proof. $\mathcal{K}^t$ is the only subset of $\mathcal{K}^t$ that has positive probability wrt. $P_{\mathcal{K}^t}$ and in fact $P_{\mathcal{K}^t}(\mathcal{K}^t) = 1$. As $\mathcal{K}^t \downarrow = \mathcal{K}$ the claim follows by definition. □

The above result shows that a probabilistic augmentation faithfully extends its base logic.

Our next result pertains to robustness of inference wrt. changes of probabilities. For example, if $\phi : 0.7$ is an inference of a knowledge base $\mathcal{K} \cup \{\psi : 0.4\}$ then we expect that $\mathcal{K} \cup \{\psi : 0.4001\} \widehat{\models} \phi : x$ with x being "close" to 0.7. In fact, probabilistic reasoning is continuous in this aspect.

Theorem 3 (Continuity)**.** *Let $\mathcal{Z}(\mathcal{L}) = (\widehat{\mathcal{W}}, \widehat{\mathcal{V}}, \widehat{\models})$ be some probabilistic augmentation and let $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ be some knowledge base, $\phi, \psi \in \mathcal{V}$ some formulas. Let $p_1, p_2, \ldots$ be a sequence with $p_i \in [0, 1]$ for all $i \in \mathbb{N}$ such that $\lim_{i\to\infty} p_i = p$. Then*

$$\lim_{i\to\infty} \left(\Pi_{\mathcal{K}\cup\{\phi:p_i\}}(\psi)\right) = \Pi_{\mathcal{K}\cup\{\phi:p\}}(\psi)$$

Proof. We have

$$\begin{aligned}
& \lim_{i\to\infty} \left(\Pi_{\mathcal{K}\cup\{\phi:p_i\}}(\psi)\right) \\
= \; & \lim_{i\to\infty} \left(p_i \Pi_{\mathcal{K}\cup\{\phi:1\}}(\psi) + (1-p_i)\Pi_{\mathcal{K}}(\psi)\right) \\
= \; & \left(\lim_{i\to\infty} p_i \Pi_{\mathcal{K}\cup\{\phi:1\}}(\psi) + \lim_{i\to\infty} (1-p_i)\Pi_{\mathcal{K}}(\psi)\right) \\
= \; & \left(p \Pi_{\mathcal{K}\cup\{\phi:1\}}(\psi) + (1-p)\Pi_{\mathcal{K}}(\psi)\right) \\
= \; & \Pi_{\mathcal{K}\cup\{\phi:p\}}(\psi)
\end{aligned}$$

□

We continue with another notion of robustness pertaining to addition of irrelevant knowledge. For that we need some further notation.

Definition 5. *The* signature $\Sigma(\mathcal{L})$ *of a logic* $\mathcal{L} = (\mathcal{W}, \mathcal{V}, \models)$ *is the set of all vocabulary elements appearing in formulas of* $\mathcal{W}$.

The above definition is a bit informal, but here are concrete examples for the logics under consideration:

- The signature of a propositional logic $\mathcal{L}_P(\mathsf{At}) = (\mathcal{W}_P(\mathsf{At}), \mathcal{V}_P(\mathsf{At}), \models_P)$ is the set of atoms: $\Sigma(\mathcal{L}_P(\mathsf{At})) = \mathsf{At}$.
- The signature of Prolog

  $$\mathcal{L}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V) = (\mathcal{W}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V), \mathcal{V}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V), \models_{\mathsf{Prolog}})$$

  is the union of predicates and variables: $\Sigma(\mathcal{L}_{\mathsf{Prolog}}(\mathsf{Pred}, U, V)) = \mathsf{Pred} \cup U$ (similar for $\mathcal{L}_{\mathsf{ASP}}$).
- The signature of abstract argumentation $\mathcal{W}_{\mathsf{AAF}}(\mathcal{A}) = \mathcal{A} \cup (\mathcal{A} \times \mathcal{A})$ is the set of arguments: $\Sigma(\mathcal{W}_{\mathsf{AAF}}(\mathcal{A})) = \mathcal{A}$.

The signature $\Sigma(\mathcal{K})$ of a knowledge base $\mathcal{K} \subseteq \mathcal{W}$ of an arbitrary logic $\mathcal{L} = (\mathcal{W}, \mathcal{V}, \models)$ is the set of vocabulary elements appearing in the knowledge base only.

The next result shows that adding independent information to a knowledge base does not change previous inferences.

Theorem 4 (Independence)**.** *Let* $\mathcal{Z}(\mathcal{L}) = (\widehat{\mathcal{W}}, \widehat{\mathcal{V}}, \widehat{\models})$ *be some probabilistic augmentation and let* $\mathcal{K}_1, \mathcal{K}_2 \subseteq \widehat{\mathcal{W}}$ *be knowledge bases with*

1. $\Sigma(\mathcal{K}_1) \cap \Sigma(\mathcal{K}_2) = \emptyset$ *and*
2. *for every* $\mathcal{K}' \subseteq \mathcal{K}_2$, $\mathcal{K}' \downarrow$ *is* $\models$*-consistent.*

Then for all formulas $\phi : p$ *such that* $\Sigma(\{\phi\}) \cap \Sigma(\mathcal{K}_2) = \emptyset$, $\mathcal{K}_1 \widehat{\models} \phi : p$ *iff* $\mathcal{K}_1 \cup \mathcal{K}_2 \widehat{\models} \phi : p$.

Proof (Sketch). Let $\mathcal{K}_1 \widehat{\models} \phi : p$ and note that for every subset $\mathcal{K}' \subseteq \mathcal{K}_2$, $\mathcal{K}' \downarrow \not\models \phi$ as $\mathcal{K}'$ is $\models$-consistent and has a disjoint signature from ϕ. The claim follows by induction on the number of elements in $\mathcal{K}_2$. Consider any $\psi : q \in \mathcal{K}_2$ and

$$\Pi_{\mathcal{K}_1 \cup \{\psi : q\}}(\phi) = q \Pi_{\mathcal{K}_1 \cup \{\psi : 1\}}(\phi) + (1 - q)\Pi_{\mathcal{K}_1}(\phi)$$

Now $\Pi_{\mathcal{K}_1 \cup \{\psi : 1\}}(\phi) = \Pi_{\mathcal{K}_1}(\phi)$ as $\{\psi : 1\}$ is consistent and has a disjoint signature from $\mathcal{K}_1$. It follows

$$\Pi_{\mathcal{K}_1 \cup \{\psi : q\}}(\phi) = q \Pi_{\mathcal{K}_1}(\phi) + (1 - q)\Pi_{\mathcal{K}_1}(\phi) = \Pi_{\mathcal{K}_1}(\phi)$$

and inductively the claim. □

Observe that it is not sufficient to only require that $\mathcal{K}_2$ is $\models$-consistent in item 2 above: $\models$ may be a non-monotonic inference relation. Then it may be the case that $\mathcal{K}_2$ is $\models$-consistent while there is a subset of $\mathcal{K}_2$ that is $\models$-inconsistent. For example, under the answer set semantics the logic program $P = \{a, a \leftarrow \textsf{not}\ a\}$ is $\models_{\textsf{ASP}}$-consistent while its subset $P' = \{a \leftarrow \textsf{not}\ a\}$ is $\models_{\textsf{ASP}}$-inconsistent

For our next results, we consider a certain (reasonable) class of base logics, i. e., those where the inference relation is reflexive. More formally, an inference relation $\models$ is *reflexive* if $\mathcal{K} \models \phi$ for all $\phi \in \mathcal{K} \cap \mathcal{V}$. Reflexivity of the inference relation of a base logic provides certain guarantees on the degrees of belief of its probabilistic augmentation.

Theorem 5 (Reflexivity)**.** *Let $\mathcal{Z}(\mathcal{L}) = (\widehat{\mathcal{W}}, \mathcal{V}, \widehat{\models})$ be some probabilistic augmentation and let $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ be some knowledge base. If $\models$ is reflexive then for every $\phi : p \in \mathcal{K}$ with $\phi \in \mathcal{V}$, $\Pi_{\mathcal{K}}(\phi) \geq p$.*

Proof. Observe that if $\mathcal{K}' = \mathcal{K} \cup \{\phi : p\}$ then $\Pi_{\mathcal{K}'}(\phi) = p\Pi_{\mathcal{K}\cup\{\phi:1\}}(\phi) + (1 - p)\Pi_{\mathcal{K}}(\phi)$. As $\Pi_{\mathcal{K}\cup\{\phi:1\}}(\phi) = 1$ due to the reflexivity of $\models$, we have $\Pi_{\mathcal{K}'}(\phi) \geq p$. □

Theorem 6 (Strengthening)**.** *Let $\mathcal{Z}(\mathcal{L}) = (\widehat{\mathcal{W}}, \mathcal{V}, \widehat{\models})$ be some probabilistic augmentation and let $\mathcal{K} \subseteq \widehat{\mathcal{W}}$ be some knowledge base and $\phi \in \mathcal{V}$. If $\models$ is reflexive then $\Pi_{\mathcal{K}}(\phi) \leq \Pi_{\mathcal{K}\cup\{\phi:p\}}(\phi)$ for every $p > 0$. If, in addition $\Pi_{\mathcal{K}}(\phi) < 1$ then $\Pi_{\mathcal{K}}(\phi) < \Pi_{\mathcal{K}\cup\{\phi:p\}}(\phi)$.*

Proof. For the first claim, observe

$$\begin{aligned}\Pi_{\mathcal{K}\cup\{\phi:p\}}(\phi) &= p\Pi_{\mathcal{K}\cup\{\phi:1\}}(\phi) + (1-p)\Pi_{\mathcal{K}}(\phi)\\ &= p\Pi_{\mathcal{K}\cup\{\phi:1\}}(\phi) + \Pi_{\mathcal{K}}(\phi) - p\Pi_{\mathcal{K}}(\phi)\end{aligned}$$

and $\Pi_{\mathcal{K}\cup\{\phi:1\}}(\phi) = 1$ as $\models$ is reflexive. We get

$$\begin{aligned}\Pi_{\mathcal{K}\cup\{\phi:p\}}(\phi) &= p + \Pi_{\mathcal{K}}(\phi) - p\Pi_{\mathcal{K}}(\phi)\\ &= \Pi_{\mathcal{K}}(\phi) + \underbrace{p(1 - \Pi_{\mathcal{K}}(\phi))}_{\geq 0}\end{aligned}$$

showing $\Pi_{\mathcal{K}}(\phi) \leq \Pi_{\mathcal{K}\cup\{\phi:p\}}(\phi)$ for every $p > 0$ and $\Pi_{\mathcal{K}}(\phi) < \Pi_{\mathcal{K}\cup\{\phi:p\}}(\phi)$ for $\Pi_{\mathcal{K}}(\phi) < 1$. □

Let us close our analysis with some brief comments regarding to computational complexity and algorithms. In its generality, computing the exact value $p \in [0,1]$ s. t. $\mathcal{K}\widehat{\models}\phi : p$ for some probabilistically augmented knowledge base $\mathcal{K}$ and formula ϕ is an intractable problem. A naïve algorithm for computing p would consider all subsets of $\mathcal{K}$, check whether $\mathcal{K}\downarrow$ entails ϕ w. r. t. the underlying entailment relation $\models$, and then accumulate the probabilities of all these subsets. Even if the problem $\mathcal{K}\downarrow \models \phi$ is decidable in polynomial time, by considering all subsets of $\mathcal{K}$, we need exponential total time. However, there are ways to avoid this complexity. For one, we can resort to approximating p instead of computing p exactly. In fact, the Monte-Carlo method has already been shown to be successful for the special cases of ProbLog [12] and the constellation approach to abstract argumentation [35]. Algorithm 1 shows a general Monte-Carlo approach to approximate p in $\mathcal{K}\widehat{\models}\phi : p$. Given some integer $N \in \mathbb{N}$ it samples N subsets of $\mathcal{K}$ w. r. t. to their probability (lines 3–7) and checks whether ϕ is entailed in that subset (line 8). It then accumulates positive cases and returns the average value (lines 9–10). Assuming that the random number generator in line 5 is truly random (uniform distribution), then by the law of large numbers we get the following general result.

Theorem 7. *For $\mathcal{K}\widehat{\models}\phi : p$, $\lim_{N\to\infty} MC(\mathcal{K}, \phi, N) = p$.*

The main complexity of Algorithm 1 comes from the entailment test in line 8, which has to be executed N times.

For some cases, approximation may not be necessary and an efficient exact algorithm for computing p in $\mathcal{K}\widehat{\models}\phi : p$ can be devised after all. For example, in [20] it is shown that computing p in $\mathcal{K}\widehat{\models}^{\mathsf{ST}}_{\mathsf{AAF}}\mathbf{a} : p$—i. e., computing the probability p of an argument **a** in the constellation approach to abstract argumentation w. r. t. stable semantics—can be done in polynomial time. Investigating such cases in our general setting is left for future work.

5. Novel probabilistic augmentations

In the following, we discuss two novel instantiations of our framework.

5.1. The probabilistic augmentation of classical propositional logic

Extending classical logic with probabilistic reasoning capabilities has a long tradition in KR [41, 27]. While, syntactically, the probabilistic augmentation of propositional logic $\mathcal{Z}(\mathcal{L}_P(\mathsf{At}))$ is a classical Nilsson-style probabilistic logic

Algorithm 1 $MC(\mathcal{K}, \phi, N)$: Approximating p in $\mathcal{K} \widehat{\models} \phi : p$ using the Monte-Carlo method

```
Input:   knowledge base $\mathcal{K}$, formula $\phi$, $N \in \mathbb{N}$
Output:  $p$

$psum = 0$
for $i = 1, \ldots, N$ do
    $\mathcal{K}' \leftarrow \emptyset$
    for $\psi : d \in \mathcal{K}$ do
        $q \leftarrow$random number in $[0, 1)$
        if $d > q$ then
            $\mathcal{K}' \leftarrow \mathcal{K}' \cup \{\psi\}$
    if $\mathcal{K}' \models \phi$ then
        $psum \leftarrow psum + 1$
return $psum/N$
```

[41], our semantics seems not to have been investigated in this context as classical Nilsson-style probabilistic logics usually define its probabilistic semantics by considering probability functions on interpretations.

Let us consider an example that illustrates the probabilistic augmentation of propositional logic.

Example 9. *Let us extend Example 1 by considering now the following knowledge base $\mathcal{K}$:*

$$\begin{array}{ll} \mathsf{sc} \Rightarrow \mathsf{sp} & : 0.6 \\ \mathsf{ss} \Rightarrow \mathsf{sp} & : 0.6 \\ \mathsf{us} \Rightarrow \mathsf{sp} \vee \mathsf{jo} & : 0.9 \\ \mathsf{sc} & : 0.7 \\ \mathsf{ss} & : 0.6 \end{array}$$

where sc means “suspicious content”, ss means “suspicious subject”, sp means “spam”, us means “unknown sender”, and jo means “job offer”. For example, now the third formula can then be read as “a mail from an unknown sender is either spam or a job offer with probability 0.9”. Consider the subset $\mathcal{K}' \subseteq \mathcal{K}$ with

$$\mathcal{K}' = \{\mathsf{sc} \Rightarrow \mathsf{sp} : 0.6, \mathsf{sc} : 0.7\}$$

and verify $P_{\mathcal{K}}(\mathcal{K}') = 0.00672$ *and* $\mathcal{K}' \downarrow \models_P$ $\textsf{\textit{sp}}$*. In fact, we obtain*

$$\mathcal{K}\widehat{\models_P}\textsf{\textit{sp}} : 0.6288$$

We leave a deeper investigation of $\mathcal{Z}(\mathcal{L}_P(\mathsf{At}))$ for future work.

5.2. The probabilistic augmentation of answer set programming

To our knowledge, the only attempt to extend answer set programming with probabilities is P-log [2], a declarative language capable of reasoning which combines both logical and probabilistic arguments. Instead, in this section let us consider the probabilistic augmentation of an answer set program $\mathcal{Z}(\mathcal{L}_{\mathsf{ASP}}(\mathsf{Pred}, U, V))$—syntactically analogous to ProbLog programs—and let us illustrate its semantics by the means of an example.[7]

Example 10. *Let us augment with probabilities the answer set program introduced in Example 3:*

$$\begin{array}{ll} \textsf{\textit{drill}} \leftarrow \textsf{\textit{alarm}}, \textsf{\textit{not real}} & : 0.2 \\ \textsf{\textit{real}} \leftarrow \textsf{\textit{alarm}}, \textsf{\textit{not drill}} & : 0.9 \\ \textsf{\textit{alarm}} & : 1 \end{array}$$

Hence, $\mathcal{K} \widehat{\models_{\textsf{\textit{ASP}}}} \textsf{\textit{real}} : 0.72$ *as the only subset* $\mathcal{K}' \models_{\textsf{\textit{ASP}}} \textsf{\textit{real}}$ *is* $\mathcal{K}' = \{\textsf{\textit{r}} \leftarrow \textsf{\textit{a}}, \textsf{\textit{not d}}; \textsf{\textit{a}}\}$.

We leave a deeper investigation of $\mathcal{Z}(\mathcal{L}_{\mathsf{ASP}}(\mathsf{Pred}, U, V))$ for future work.

6. Related work

Our approach addresses a very general topic—namely the combination of logic and probability—that has been addressed before in one way or the other by many researchers. Halpern and colleagues [27, 17, 1] combine classical first-order logic with different probabilistic interpretations, and [19] in the case of abstract argumentation. Their focus is similar in spirit to classical Nilsson-style probabilistic logic [41] and they discuss probability distributions over the interpretations of

[7] Although no formal definitions or analysis are provided, at first sight it appears that part of the approach discussed in [15] is close in spirit to the probabilistic augmentation of an answer set program we propose in this paper.

the language, not over subsets of knowledge bases as we do. In particular, they address the challenge of considering first-order logic instead of propositional logic and the computational issues raised by, e. g., considerations of infinity. Furthermore, Wilson [58] pointed out the possibility to augment a logic with probabilities in a general way. However, he discusses this idea in a very abstract way without technical details. Of particular interest is the case of concurrent constraint programming [54], and its probabilistic extensions, e.g. [13, 25], as they can become another interesting test-case between our approach and ad-hoc proposals, as it is in the case of our proposed probabilistic version of ASP (cf. Section 5.2) versus the P-log [2] proposal.

Modern approaches for combining logic and probability can be found within the field of *statistical relational AI* [10], which are general approaches to deal with relational information (i. e., limited first-order logic expressions) and probabilistic reasoning. For example, Markov logic [52] is an extension of first-order logic with weights on formulas. These weights are used to obtain probabilities on formulas using a log-linear probabilistic model over the interpretations. This gives a robust knowledge representation formalism which is apt to be used as the output for a variety of machine learning tasks, see e. g. [14].

All the approaches mentioned so far consider a different, somewhat Bayesian point of view in the combination of logic and probability than we do here. The above approaches use probabilities over interpretations do obtain degrees of belief in inferred information. In our work, we take a frequentist approach as interpret probabilities over formulas s. t. the probability of a formula corresponds to its frequency of actually occurring in the knowledge base. Both points of view provide important insights into the general challenge of combining logic and probability. In addition, as already noted in [55], probabilistic facts (or binary switches) are expressive enough to represent a wide range of models including Bayesian networks.

Specific instances of our general framework have been developed and investigated before, some of them have already been discussed in Sections 3.1 and 3.2. One of those first approaches is the distribution semantics for logic programs [55], which is also the foundation for ProbLog as discussed in Section 3.1. However, in the original proposal [55] only facts of logic programs were annotated with probabilities. Still, the general idea of considering all possible subsets of the set of facts and weighing the inferences by the probabilities of these subsets, can be found there as well. The *independent choice logic* and its predecessor *probabilistic Horn abduction* [45, 46, 47] feature a similar setting where facts of logic programs are treated as probabilistic hypotheses. Kohlas and colleagues [33, 32, 31] also consider combinations of logics with probabilities using the subset-based interpretation. Their focus is on classical logics, though, and algorithmic issues for reasoning. They do not consider conceptual questions and general logic instantiations. Probabilistic data bases [57] are another instance of our general framework. There, tuples of a relational data base instance can be annotated with probabilities, which model the likelihood that the tuple is actually present. Conceptually, this is a probabilistic augmentation on a restricted first-order logic where probabilities smaller than 1 are assigned to ground atoms. One of the driving research challenges is probabilistic data bases is, however, algorithmic reasoning mechanisms [9].

The aim of this paper is to provide a general umbrella that unifies these works from different areas. To the best of our knowledge, no other work has considered the issue of probabilistic augmentation in the generality proposed in this paper.

7. Discussion and summary

In this paper we developed a general scheme for adding probabilistic reasoning capabilities to any knowledge representation formalism. Pivotal in our proposal is the notion of probabilistic augmentation of a knowledge representation formalism, which extends it by enabling probabilities to be expressed on the logical formulas of the chosen formalism. In addition to showing that it subsumes existing approaches, we provided an extensive analysis which includes proofs of desirable behaviours, such as trivialisation, continuity, independence, reflexively, and strengthening. We also showed how novel instantiations of our approach can be derived using propositional logic and answer set programming as examples.

Our logical setting is general enough to use a wide variety of logics as a base logic. In addition to our examples, the use of e.g. modal logics, default logics, epistemic logics, temporal logics, paraconsistent logics, and others is straightforward as long as they can be cast into our general form $\mathcal{L} = (\mathcal{W}, \mathcal{V}, \models)$. Furthermore, as a probabilistic augmentation $\mathcal{Z}(\mathcal{L})$ of a logic $\mathcal{L}$ is a logic itself it can probabilistically augmented as well, yielding a doubly-probabilistically augmented logic $\mathcal{Z}(\mathcal{Z}(\mathcal{L}))$. While the practical use of the latter may be disputable, it shows that we have a rich framework with high expressivity.

The proposal by [28] of considering the constellation approach to argumentation to the case of extended argumentation frameworks—i.e. argumentation frameworks that allow attacks to be the target of attacks themselves—can be encompassed as another special case of our approach. In addition, extending it to the case of other extended argumentation frameworks will be the subject of future investigation. Also the logics of [44], [50, 18] and [53] can be represented as probabilistic augmentations of the their corresponding base logics *Abstract Dialectical Frameworks* [5], *Bipolar Abstract Argumentation Frameworks* [6] and ASPIC$^+$ [40].

We also showed that, despite the fact that our framework relies on probabilistic independence assumptions between formulas, we are able to model probabilistic dependencies as well (see Section 3.2). The results from this section can easily be applied to other probabilistic models with dependency features such as Bayesian Networks [43]. Finally, we agree that relaxing the probabilistic independence assumptions between formulas is important, and we aim at doing so in general terms as part of future work. Indeed, it would be interesting to explore how to generalise the framework to capture other probabilistic logics based on distribution semantics with no probabilistic independence such as Lukasiewicz's Probabilistic Logic Programming (e. g., [37, 38]).[8] In this way we will have a solid, coherent setting to use for theoretical and experimental comparison with other approaches relaxing the independence assumption such as [36, 55, 48].

This paper lays the foundation for a general approach to probabilistic reasoning that has the potential to create synergies between different fields interested in incorporating probability into a specific framework. For example, fields such as probabilistic data bases (see previous section) have developed highly efficient procedures for reasoning problems and our framework allows for lifting these ideas and applying them to other formalisms, such as the constellation approach to abstract argumentation. The exploitation of our framework in these matters is part of ongoing work.

Acknowledgements The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (grant KE 1686/3-1).

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

[8] We thank one of the anonymous reviewers for this suggestion.

[1] Fahiem Bacchus, Adam J. Grove, Joseph Y. Halpern, and Daphne Koller. From Statistical Knowledge Bases to Degrees of Belief. *Artificial Intelligence*, 87(1-2):75–143, 1996.

[2] Chitta Baral, Michael Gelfond, and Nelson Rushton. Probabilistic Reasoning With Answer Sets. In Vladimir Lifschitz and Ilkka Niemelä, editors, *Logic Programming and Nonmonotonic Reasoning*, pages 21–33. Springer Berlin Heidelberg, 2004.

[3] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, 2004.

[4] Ivan Bratko. *Prolog programming for artificial intelligence*. Addison Wesley, 2001.

[5] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Peter Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 803–809, 2013.

[6] C. Cayrol and M. C. Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In Lluís Godo, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 378–389, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[7] Federico Cerutti and Matthias Thimm. A General Approach to Reasoning with Probabilities (Extended Abstract). In *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning (KR'19)*, October 2018.

[8] Federico Cerutti and Matthias Thimm. Probabilistic Augmentations for Knowledge Representation Formalisms. In *Proceedings of the Workshop on Hybrid Reasoning and Learning (HRL'18)*, October 2018.

[9] Nilesh Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *The International Journal on Very Large Data Bases*, 16(4):523–544, 2007.

[10] Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2016.

[11] Luc De Raedt and Angelika Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100(1):5–47, 2015.

[12] Luc De Raedt, Angelika Kimmig, Bernd Gutmann, Kristian Kersting, Vitor Santos Costa, and Hannu Toivonen. Probabilistic inductive querying using problog. Technical Report CW 552, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, June 2009.

[13] Alessandra Di Pierro and Herbert Wiklicky. Probabilistic concurrent constraint programming: Towards a fully abstract model. In Luboš Brim, Jozef Gruska, and Jiří Zlatuška, editors, *Mathematical Foundations of Computer Science 1998*, pages 446–455, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[14] Pedro Domingos and Daniel Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool, San Rafael, CA, 2009.

[15] Stanislav Dragiev, Alessandra Russo, Krysia Broda, Mark Law, and Calin-Rares Turliuc. An abductive-inductive algorithm for probabilistic inductive logic programming. In *Proceedings of the 26th International Conference on Inductive Logic Programming (Short papers)*, pages 20–26, 2016.

[16] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[17] Ronald Fagin and Joseph Y. Halpern. Reasoning about knowledge and probability. *Journal of the ACM*, 41(2):340–367, 1994.

[18] Bettina Fazzinga, Sergio Flesca, and Filippo Furfaro. Probabilistic bipolar abstract argumentation frameworks: complexity results. In *IJCAI*, pages 1803–1809, 2018.

[19] Bettina Fazzinga, Sergio Flesca, and Filippo Furfaro. Complexity of fundamental problems in probabilistic abstract argumentation: Beyond independence. *Artificial Intelligence*, 268:1 – 29, 2019.

[20] Bettina Fazzinga, Sergio Flesca, and Francesco Parisi. On the complexity of probabilistic abstract argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, 2013.

[21] Norbert Fuhr. Probabilistic datalog: Implementing logical information retrieval for advanced applications. *Journal of the Association for Information Science and Technology*, 51(2):95–110, 2000.

[22] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Morgan & Claypool Publishers, 2012.

[23] Michael Gelfond and Nicola Leone. Logic programming and knowledge representation – the A-Prolog perspective. *Artificial Intelligence*, 138(1–2):3–38, 2002.

[24] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.

[25] Vineet Gupta, Radha Jagadeesan, and Vijay Saraswat. Probabilistic concurrent constraint programming. In Antoni Mazurkiewicz and Józef Winkowski, editors, *CONCUR '97: Concurrency Theory*, pages 243–257, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[26] Petr Hájek. *Metamathematics of Fuzzy Logic*. Dordrecht: Kluwer, 1998.

[27] Joseph Y. Halpern. An Analysis of First-Order Logics of Probability. *Artificial Intelligence*, 46:311–350, 1990.

[28] Nguyen Duy Hung. The distribution semantics of extended argumentation. In *Proceedings of the 18th International Symposium on Knowledge and Systems Sciences*, pages 197–211, 2017.

[29] Anthony Hunter. Some foundations for probabilistic abstract argumentation. In *Proceedings of the 4th International Conference on Computational Models of Argument*, pages 117–128, 2012.

[30] Anthony Hunter. Probabilistic qualification of attack in abstract argumentation. *International Journal of Approximate Reasoning*, 55(2):607–638, 2014.

[31] Jürg Kohlas. Probabilistic argumentation systems: A new way to combine logic with probability. *Journal of Applied Logic*, 1(3):225 – 253, 2003. Combining Probability and Logic.

[32] Jürg Kohlas, Bernhard Anrig, Rolf Haenni, and Paul-André Monney. Model-based diagnostics and probabilistic assumption-based reasoning. *Artificial Intelligence*, 104(1-2):71–106, September 1998.

[33] Jürg Kohlas and Paul-André Monney. *Probabilistic Assumption-Based Reasoning*, pages 82–135. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

[34] Robert A. Kowalski. The early years of logic programming. *Communications of the ACM*, 31(1):38–43, 1988.

[35] Hengfei Li, Nir Oren, and Timothy J. Norman. Probabilistic argumentation frameworks. In *Proceedings of the First International Workshop on the Theory and Applications of Formal Argumentation*, pages 1–16, 2011.

[36] Hengfei Li, Nir Oren, and Timothy J Norman. Relaxing independence assumptions in probabilistic argumentation. In *Tenth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2013)*, 2013.

[37] Thomas Lukasiewicz. Probabilistic logic programming. In *ECAI*, pages 388–392, 1998.

[38] Thomas Lukasiewicz. Local probabilistic deduction from taxonomic and probabilistic knowledge-bases over conjunctive events. *International Journal of Approximate Reasoning*, 21(1):23–61, 1999.

[39] Theofrastos Mantadelis and Gerda Janssens. Dedicated Tabling for a Probabilistic Setting. In Manuel Hermenegildo and Torsten Schaub, editors, *Technical Communications of the 26th International Conference on Logic Programming*, volume 7 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 124–133, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[40] Sanjay Modgil and Henry Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5:31–62, 2014.

[41] Nils J. Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

[42] Jeff B. Paris. *The Uncertain Reasoner's Companion – A Mathematical Perspective*. Cambridge University Press, 1994.

[43] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Springer-Verlag, 1988.

[44] Sylwia Polberg and Dragan Doder. Probabilistic abstract dialectical frameworks. In *Proceedings of the 14th European Conference on Logics in Artificial Intelligence*, pages 591–599, 2014.

[45] David Poole. Representing diagnostic knowledge for probabilistic horn abduction. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'91, pages 1129–1135, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[46] David Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1):81 – 129, 1993.

[47] David Poole. The independent choice logic and beyond. In Luc De Raedt, Paolo Frasconi, Kristian Kersting, and Stephen Muggleton, editors, *Probabilistic Inductive Logic Programming: Theory and Application*, number 4911 in Lecture Notes in Artificial Intelligence. Springer, 2008.

[48] David Poole. Probabilistic programming languages: Independent choices and deterministic systems. In *Heuristics, Probability and Causality: A Tribute to Judea Pearl*. College Publications, 2010.

[49] Nico Potyka and Matthias Thimm. Consolidation of probabilistic knowledge bases by inconsistency minimization. In *Proceedings of the 21st European Conference on Artificial Intelligence*, pages 729–734, 2014.

[50] Carlo Proietti. Polarization and bipolar probabilistic argumentation frameworks. In *AIˆ 3@ AI* IA*, pages 22–27, 2017.

[51] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[52] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1–2):107–136, 2006.

[53] Tjitze Rienstra. Towards a probabilistic Dung-style argumentation system. In *Proceedings of the First International Conference on Agreement Technologies*, pages 138–152, 2012.

[54] Vijay A. Saraswat and Martin Rinard. Concurrent constraint programming. In *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '90, pages 232–245, New York, NY, USA, 1990. ACM.

[55] Taisuke Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming*, pages 715–729, 1995.

[56] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[57] Dan Suciu, Dan Olteanu, Christopher Re, and Christoph Koch. *Probabilistic Databases*. Morgan & Claypool Publishers, 2011.

[58] Nic Wilson. How much do you believe? *International Journal of Approximate Reasoning*, 6(3):345–365, May 1992.